Model Predictive Controller for Autonomous Vehicle Guidance

Mayukh Sattiraju
Department of Electrical and Computer Engineering, Clemson
msattir@clemson.edu

Sai Prasant P Velusamy
Department of Automotive Engineering
CU-ICAR
svelusa@clemson.edu

Pengyu Zhao
Department of Automotive Engineering
CU-ICAR
pengyuz@clemson.edu

***Abstract*—A Model Predictive Control (MPC) approach for controlling the guidance of an autonomous vehicle is presented.We formulate a predictive control problem in order to best follow a given path by controlling the front steering angle. while fulfilling various physical and design constraints. At each time step a trajectory is assumed to be known over a finite horizon, and an MPC controller computes the system inputs in order to best follow the desired trajectory at a given speed. We formulate the model based on various assumptions and its analysis are discussed in this project**

I. Introduction

Model predictive controllers rely on dynamic models of the process, most often linear empirical models obtained by system identification. The main advantage of MPC is the fact that it allows the current timeslot to be optimized, while keeping future timeslots in account. MPC has the ability to anticipate future events and can take control actions accordingly. PID and LQR controllers do not have this predictive ability.

The idea how we use this approach can be explained as the driver looks at the road ahead of him and taking into account the present state and the previous action predicts his action up to some distance ahead, which we refer to as the prediction horizon. Based on the prediction, the driver adjusts the driving direction. The same is done by a Model Predictive Controller in place of a driver. This paper focus on an autonomous vehicle trajectory guidance problem.

In Model Predictive Control (MPC) a model of the plant is used to predict the future evolution of the system. Based on this prediction, at each time step t a performance index is optimized under operating constraints with respect to a sequence of future input moves in order to best follow a given trajectory. The first of such optimal moves is the control action applied to the plant at time t. At time t + 1, a new optimization is solved over a shifted prediction horizon.

In this paper, we are trying to realize the trajectory tracking and by applying the MPC under 3 different model systems. We observed the performance of trajectory based on different vehicle status and controller parameters. The simulation results are clearly showing the changes on the lag time and outputs. And out team also applied the boundary constraints of steer angle and lateral deviation of the vehicle which results the different trajectory performances on the positon and steering.

II. Prior Art

The trend focus on the autonomous vehicle guidance involves the vehicle control in a tractor by 3D trajectory tracking and 3D trajectory planning to avoid any obstacle in the path. There are many ways to model a vehicle which involve a bicycle model or constructing a particle motion of a vehicle along a curved path in a Frenet frame[1] and extracting the equations of motion for non-linear behavior and deriving the hard and soft constraints controlling the motion. The advantage of using this model is that, it considers on the velocity, acceleration and its relation with the orientation and offset with the reference trajectory as parameters.The problem falls under the optimal control minimizing the cost function of the control parameters of the vehicles bound by the constraints over the prediction horizon

The difficulty in implementing a Model predictive controller is the computational capability required in solving the nonlinearity of the states and constraints considered and the optimization of cost function needs to be iterated with the weights of the control parameters in the cost function. The constraints are bound to control the cost function to avoid an overshoot or long settling time.

III. Problem Definition

To control the trajectory guidance of an autonomous vehicle in its motion with reference to an existing trajectory by a defined distance and orientation by modeling and implementation of a Model Predictive controller.

This paper implements MPC for trajectory tracking for 3 different models of increasing complexity. The first, is a linear Bicycle Model. The second, is a Bicycle Model which has the error measurements as states. And third, is a Particle Model with error measurements as states. For case two and three, the nonlinear models are linearized by making particular assumptions, then these linearized Models are used by the MPC to follow the reference path.

IV. Vehicle Models and Coordinate Systems

A. Coordinate Systems

In this paper, we are mainly focus on the development of a trajectory guidance for vehicles under the environment of driving on a highway. The control purpose is to keep the vehicle in its lane. And the position of the vehicle can be measured with observers like sensors/camera.

For the basic coordinate system of the vehicles, we assume the road can be described in a global coordinate frame[3]

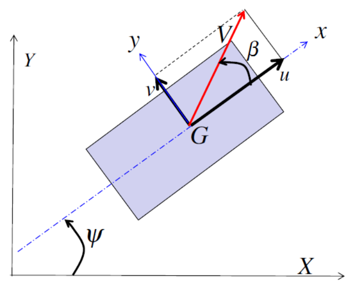

Fig. 1. Coordinate System in Body Coordinates

The equations of motions for this vehicle model are:

$$m\ddot{X} = \sum F_x$$
$$m\ddot{Y} = \sum F_y$$
$$J_z\ddot{\psi} = \sum M_z$$

B. Bicycle Model

For easy to analysis the problem, we simplified the vehicle at each axis through a bicycle model. And we assume the speed v_x in the longitudinal direction can be considered as a constant every time instant.

In this model, the wheels on each side are lumped into a single wheel so that a simple linear model and small approximation can be applied due to the low speed.

The Equations of Motion become:

$$m(\dot{V} - rv) = \sum F_x$$
$$m(\dot{v} + rV) = \sum F_y$$
$$J_z\dot{r} = \sum M_z$$

In terms of the slip angle BETA, the equations can be re-written as:

$$m(\dot{V} - rV\beta) = \sum F_x$$
$$mV(\dot{\beta} + rV) = \sum F_y$$
$$J_z\dot{r} = \sum M_z$$

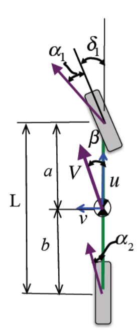

Fig. 2. Bicycle Model

The axle slip angles are defined as:

$$\alpha_1 = \beta + \frac{r}{V}a - \delta_1$$
$$\alpha_2 = \beta - \frac{r}{V}b - \delta_2$$

1) Linearized Bicycle Model: The vehicle or plant model is derived out as a bicycle model in which both the wheels in an axle are lumped together to form a single element. At low speeds, the influence of external forces and moments are ignored for the purpose of creating a linearized bi-cycle model.

The equations for the lateral acceleration and longitudinal acceleration for a steer angle are given as[2]

$$mV(\dot{\beta} + r) = -(C1 + C2)\beta - \frac{1}{V}(aC1 - bC2)r + C1\delta + F_{ye}$$
$$J_z\dot{r} = (-aC1 + bC2)\beta + \frac{1}{V}(-a^2C1 - b^2C2)r + aC1\delta + M_{ze}$$

C. Particle Model

A reduced vehicle model - such as the Particle Model can be computationally very efficient in reducing the MPC processing time. Here we also explore the Particle Model to model the Autonomous vehicle, by deriving the equations of motion for this model, then defining error terms for Trajectory Guidance and build an augmented equation system that describe the motion.

The Particle Model is depicted by Fig 4. For this model the Equations for lateral and longitudinal motion used are[4]

$$\dot{a}_t = {}^1\!/_{T_{at}}\,(a_{t,d} - a_t)$$
$$\ddot{\psi}_p = {}^1\!/_{T_{\dot{\psi}_p}}\left(\dot{\psi}_{p,d} - \dot{\psi}_p\right)$$

The values of $a_{t,d}$ and $\dot{\psi}_{p,d}$ are inputs given to the particle model and T_{at} and $T_{\dot{\psi}_p}$ are first order time constants.

V. Environment Models

A. Bicycle Model

For solving Trajectory tracking with a non-linear bicycle model and measured errors as states. Here is a potential model. This model introduces two error measurements with respect to the reference trajectory a angular error e-Psi and a lateral error e-y.

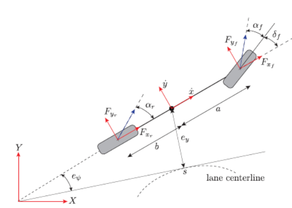

Fig. 3. Non Linear Bicycle Model with error measurements

From the diagram the equations for the measured error variables is given as

$$\dot{e}_y = v_x \sin(e_\psi) + v_y \cos(e_\psi)$$
$$\dot{e}_\psi = \dot{\psi} - ks$$
$$\dot{s} = \frac{1}{1 - ke_y}(v_x \cos(e_\psi) - v_y \sin(e_\psi))$$

B. Particle Model

The error terms in this model are defined as:

- Lateral Error - Perpendicular distance from the Particle and the reference trajectory

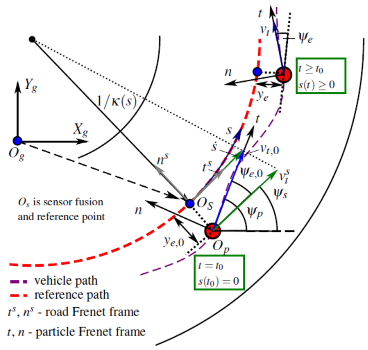

Fig. 4. Non Linear Particle Model with error measurements

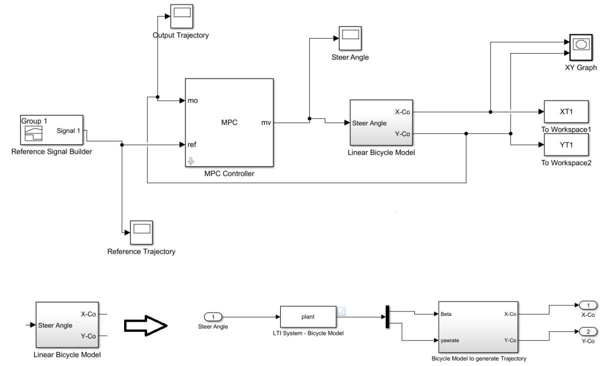

Fig. 5. Simulink Block Diagram for MPC with Bicycle Model

- Angular alignment error - Deviation from the Tangent to the particle heading direction

The equations for the alignment error (ψ) and the lateral error (y_e) are given as

$$\dot{\psi}_e = \dot{\psi}_p - V\cos(\psi_e)\left(\frac{k(s)}{1-y_e k(s)}\right)$$
$$\dot{y}_e = V\sin(\psi_e)$$

VI. Equations of Motion for Trajectory Tracking

Given these Vehicle Models and error calculations, we can formulate the problem of Trajectory Tracking. The aim of the MPC is to reduce a cost function that penalizes the error in current position as opposed to a desired position on the reference path.

A. Trajectory Tracking for Linear Bicycle Model

From the section on the Bicycle Model we recall the Equations of Motion describing the lateral and longitudinal velocity can be given by

$$\underbrace{\begin{bmatrix} \dot{v}_y(t) \\ \ddot{\psi}(t) \end{bmatrix}}_{\dot{x}(t)} = \underbrace{\begin{bmatrix} \frac{-(C1+C2)}{mV_x} & -v_x - \frac{aC1-bC2}{mV_x} \\ -\frac{aC1-bC2}{J_z V_x} & -\frac{a^2C1+b^2C2}{J_z V_x} \end{bmatrix}}_{A} \begin{bmatrix} v_y(t) \\ \dot{\psi}(t) \end{bmatrix} + \underbrace{\begin{bmatrix} \frac{C1}{m} \\ \frac{aC1}{J_z} \end{bmatrix}}_{B} \delta(t)$$

With the knowledge of $\dot{v}_y$ and $\ddot{\psi}$ the trajectory (X,Y) projected can be calculated as

$$X = \int V\left[\cos\beta\cos\psi - \sin\beta\sin\psi\right]dt$$
$$Y = \int V\left[\cos\beta\cos\psi + \sin\beta\sin\psi\right]dt$$
$$where,$$
$$\psi(t) = \int r(t)dt$$
$$V\beta = v_y$$

This generated trajectory is fed as the "Observer" variable for the MPC and the reference path is fed as the "Reference" variable for the MPC.

The MPC varies the steer angle delta in order to follow the reference trajectory.

The Simulink Block Diagram to achieve this is

B. Trajectory Tracking for Non-Linear Bicycle Model

Trajectory Guidance for a Non-linear bicycle model with the lateral and longitudinal error added as additional states. For reference paths other than paths with a constant curvature, the lateral and longitudinal error described by (some equation above) changes dynamically. Thus it is intuitive to include them as states in the vehicle model.

An augmented equations system is created by combining the bicycle models equations for lateral and longitudinal acceleration and the equations that describe the lateral and angular errors for the bicycle model. To these equations we also add two additional states, reference path variable s, and velocity.

The equations of motion for the Non-Linear Bicycle Model with error measurements as states is given by these equations. Formulated and described by [3]

$$\ddot{x} = \dot{y}\dot{\psi} + a_x$$
$$\ddot{y} = -\dot{x}\dot{\psi} + \frac{1}{m}(2F_{y,f} + 2F_{y,r})$$
$$\ddot{\psi} = \frac{1}{I_z}(2aF_{y,f} - 2bF_{y,r})$$
$$\dot{e}_\psi = \dot{\psi} - \kappa\dot{s}$$
$$\dot{e}_y = \dot{x}sin(e_\psi) + \dot{y}cos(e_\psi)$$

Fig. 6. Equations of Motion for Nonlinear Bicycle Model with Error measurements as states

The following assumptions are adopted to linearize the model.

- Angular error e_ψ is small, approximating the trigonometric functions
- The lateral velocity $\dot{y}$ and yaw rate $\dot{\psi}$ are small
- The curvature K, is known

The equations of motion for the linearized system thus, can be described as

As with the previous case the Trajectory can be computed, and fed to the MPC.

The MPC structure for this Vehicle Model is described by this Simulink Block Diagram.

$$
\begin{bmatrix} \dot{v}_x \\ \dot{v}_y \\ \ddot{\psi} \\ \dot{e}_\psi \\ \dot{e}_y \\ s \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\dfrac{C1+C2}{mV_x} & -\dfrac{aC1+bC2}{mV_x} & 0 & 0 & 0 \\ 0 & -\dfrac{aC1-bC2}{J_zV_x} & -\dfrac{a^2C1-b^2C2}{J_zV_x} & 0 & 0 & 0 \\ -K & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & V_x & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \\ e_\psi \\ e_y \\ \dot{s} \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ \dfrac{C1}{m} & 0 \\ \dfrac{aC1}{J_z} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_f & a_x \end{bmatrix}
$$

Fig. 7. Equations of Motion - Linearized Bicycle Model with error measurements as states

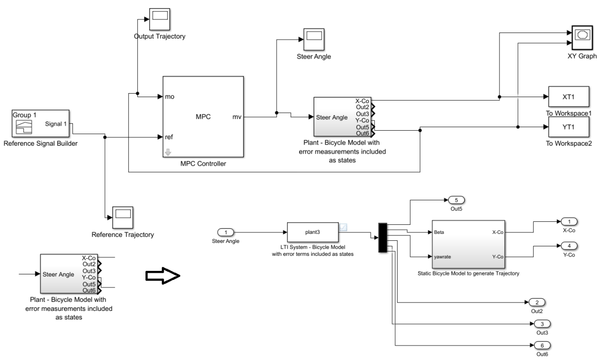

Fig. 8. Simulink Block Diagram for MPC with Bicycle Model with error measurements included as states

C. Trajectory Tracking for Particle Model

The equations of motion for the trajectory tracking of the particle Model including the alignment error and lateral error as states is given as, (as described in [2])

$$
\begin{aligned}
\dot{v}_t &= a_t, \\
\dot{\psi}_e &= \dot{\psi}_p - v_t \cos(\psi_e)\left(\frac{\kappa(s)}{1-y_e\kappa(s)}\right), \\
\dot{y}_e &= v_t \sin(\psi_e), \\
\dot{a}_t &= 1/T_{a_t}\left(a_{t,d} - a_t\right), \\
\ddot{\psi}_p &= 1/T_{\psi_p}\left(\psi_{p,d} - \psi_p\right), \\
\dot{s} &= v_t \cos(\psi_e)\left(\frac{1}{1-y_e\kappa(s)}\right).
\end{aligned}
$$

Fig. 9. Particle Model with error measurements as states - EOM

Making the following assumptions to solve the state-space equations

- $\psi_e \cong 0$. Thus, $cos(\psi_e) \cong 1$ and $sin(\psi_e) \cong \psi_e$
- As we are dealing with relatively straight paths the product of $1 - y_e k(s)$ is approximated to 1
- We consider the velocity to remain constant, thus $a_t = 0$

The equations with these assumptions are

The state-space would be

To obtain the trajectory from the above state-space we use the yaw rate $\ddot{\psi}_p$ to obtain the yaw angle ψ_d and use this angle to resolve the Velocity vector into V_x and V_y which is then integrated to obtain X,Y for trajectory estimation.

The Simulink block diagram is

$$
\begin{aligned}
\dot{v}_t &= 0 \\
\dot{\psi}_e &= \dot{\psi}_p - v_t * r \\
\dot{y}_e &= v_t \psi_e \\
\dot{a}_t &= \frac{1}{T_{at}}\left(a_{t,d}\right) \\
\ddot{\psi}_p &= \frac{1}{T_{\alpha\psi_p}}\left(\dot{\psi}_{pd} - \dot{\psi}_p\right) \\
\dot{s} &= v_t
\end{aligned}
$$

Fig. 10. Linearized Particle Model

$$
\begin{bmatrix} \dot{v}_t \\ \dot{\psi}_e \\ \dot{y}_e \\ \dot{a}_t \\ \ddot{\psi}_p \\ \dot{s} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ -r & 0 & 0 & 0 & 1 & 0 \\ 0 & v_t & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\dfrac{1}{T_{\alpha\psi_p}} & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_t \\ \psi_e \\ y_e \\ a_t \\ \dot{\psi}_p \\ s \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \dfrac{1}{T_{at}} \\ \dfrac{1}{T_{\alpha\psi_p}} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\psi}_{pd} \\ a_{td} \end{bmatrix}
$$

Fig. 11. State Space for Linearized Particle - EOM

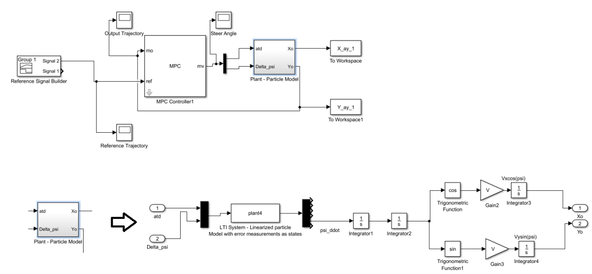

Fig. 12. Simulink Block Diagram for MPC with Particle Model with error measurements included as states

The cost function that the MPC tries to optimize in all the above three cases is given

$$
J = \sum_{k=0}^{N_p} \|y_k - r_k\|^2
$$

Where y_k is the output trajectory and r_k is the reference trajectory.

We also attempted to maintain the non-linearity of the system of equations by solving the equations non-linearly on Simulink using fsolve. The real-time simulation of the fsolve function couldn't be parsed to the simulink environment on the current version of Matlab/Simulink. Since R2017a Matlab provides a Non-Linear MPC solver that can be employed for this case.

VII. Simulations/Results

The Vehicle Models were simulated on Matlab/Simulink R2016b for trajectory guidance. The simulations were simulated for all three models by varying the Sampling Time, Prediction Horizon and the Velocity of the vehicle.

The different velocities were applied on three models. And the results show the lag due to the velocity increase. And changed the curvature only for the nonlinear bicycle

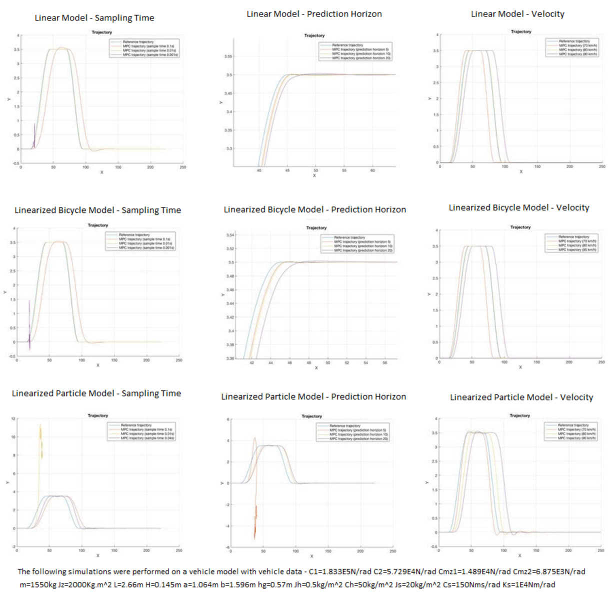

Fig. 13. Simulation for Trajectory Guidance for all 3 models and varying Sampling Time, Prediction horizon and Velocity

model. We assumed the velocity from 70 km/h to 90 km/h and observed that there was a lag increasing as the speed increasing. For the different prediction horizon, we noticed that the rise time and the overshoot were increasing with the prediction horizon rising. Also the peak time delayed by bigger prediction horizon selection. In addition, the sample time will affect the stability of the vehicle. The trajectory will have a bigger lag with bigger sample time while too small sample will cause the instability of the system. By comparing the different system models, we found that the trajectories are more sensitive to the changes of velocity, prediction horizon and sample time under the particle model.

A. *Constraints*

The manipulated and observed variable of the MPC can be constrained according to the user's needs. This paper explores constraining the MPC output - Steer Angle - Limiting the Minimum and Maximum value and also placing a constraint on the rate-of-change-of Steer Angle. Constraints on the output trajectory were are also explored, these are analogous to constraints on the lateral deviation of the particle.

These constraints were applied to the MPC using Matlab's user interface. The plots are for the linearized bicycle model.

We applied the constraints on the nonlinear model for the steering angel and the lateral deviation of the vehicle. We saw that with steering angle constraints, the trajectory became more smooth than the previous one on the corner. And it had a little bit overshoot due to the understeer angle. Then with lateral deviation constraints, the system trajectory was truncated at the boundary we set.

VIII. Conclusions and Future Work

In this paper our analysis conclude the trajectory is closed followed but not exactly as the reference trajectory provided.

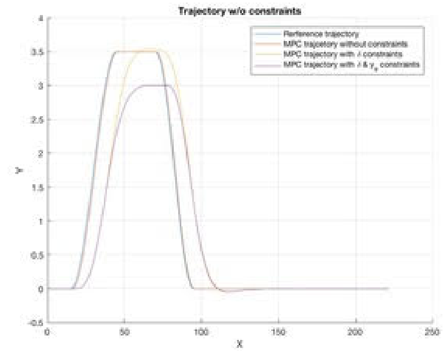

Fig. 14. Constraints applied on Steer Angle and Output Trajectory for linearized Bicycle Model

We also studied the trajectory tracking of an autonomous vehicle using MPC with bicycle model as well as a particle motion model and also attempted in solving a particle motion non linear model. We believe it is due to the assumptions and constraints associated are not robust to make it to the real case of exact tracking by the MPC. The analysis also compared the influence of speed, prection horizon and sample time on the robustness of traction tracking. The solver we used in the MPC toolbox of Simulink is not robust enough to control and tune a non-linear MPC with more constraints compared to other solvers like fmincon, ACADO.

The field of MPC is autonomous vehicle guidance has a plenty of future scope on the works which can be extended to

- Modeling dynamic obstacle detection and dynamic path planning in real time cases
- Switching to a better non linear MPC solvers like fmincon,ACADO
- Include other vehicle control parameters like torques and brake pressure
- Modelling of path planner module
- Include process noise and measurement noise as disturbance into the plant and build robust constraints
- Apply suitable filters like Kalman filters to filter out the noise and input the noise free measurements into the MPC
- Perform real time simulations in a software like CarSIM or ADAMS

References

[1] Weiskircher, Thomas, and Beshah Ayalew. "Frameworks for interfacing trajectory tracking with predictive trajectory guidance for autonomous road vehicles." American Control Conference (ACC), 2015. IEEE, 2015.
[2] Carvalho, Ashwin, et al. "Stochastic predictive control of autonomous vehicles in uncertain environments." 12th International Symposium on Advanced Vehicle Control. 2014.
[3] Course AuE 8500, Spring '17, Lecture Slides
[4] Weiskircher, Thomas, and Beshah Ayalew. "Predictive trajectory guidance for (semi-) autonomous vehicles in public traffic." American Control Conference (ACC), 2015. IEEE, 2015.